Strabon: Semantic Support for EO Data Access in TELEIOS

2nd Community Workshop
Darmstadt, 11 May 2012

Presenter: Kostis Kyzirakos (NKUA)
Outline

• The Fire Monitoring Service of NOA
• stRDF/stSPARQL and Strabon
• Examples of data and queries in the Fire Monitoring Service
• Examples of data and queries in the Virtual EO for TerraSAR-X Data
• Conclusions
The Fire Monitoring Service

Eumetsat @ 9.5° East

Back End: MonetDB / Strabon
- Corine Landcover
- Admin Boundaries
- POIs

Geospatial Ontology

Linked Geospatial Data Semantic technologies

Cataloguing Service & Metadata Creation

Processing Chain (SciQL based)

Data Vault

HotSpots

External Sources

Raw Data

Front End: GUI

Map Element

Web access based on Semantics

05/11/12
Requirements of the Fire Monitoring Service

- Need for modeling of
  - Geospatial information
  - Temporal information
  - Product metadata
  - Product content
- Need to link to other data sources
  - GIS data
  - Other information on the Web
The Data Model stRDF

- stRDF extends RDF with:
  - **Spatial literals** encoded in Well-Known Text/GML (OGC standards)
  - New Datatype for spatial literals (**strdf:geometry**)
  - **Valid time of triples** encoded by Boolean combinations of temporal constraints (ignored for the time being)
stRDF: An example
ex:BurntArea1 rdf:type noa:BurntArea.
ex:BurntArea1 noa:hasID "1"^^xsd:decimal.
ex:BurntArea1 noa:hasArea "23.7636"^^xsd:double.

ex:BurntArea1 strdf:geometry "POLYGON(( 38.16 23.7, 38.18 23.7,
38.18 23.8, 38.16 23.8, 38.16 23.7));
ex:BurntArea1 rdf:type noa:BurntArea.
ex:BurntArea1 noa:hasID "1"^^xsd:decimal.
ex:BurntArea1 noa:hasArea "23.7636"^^xsd:double.

ex:BurntArea1 strdf:geometry "<gml:Polygon
srsName='http://www.opengis.net/def/crs/EPSG/0/4121'>
<gml:outerBoundaryIs><gml:LinearRing><gml:coordinates>38.16,23.70
38.18,23.70 38.18,23.80 38.16,23.80

Spatial Data Type
Well-Known Text
Spatial Literal
Find all burnt forests close to a city

```
SELECT ?BA ?BAGEO
WHERE {
  ?R rdf:type noa:Region ;
    strdf:geometry ?R GEO ;
    noa:hasCorineLandCoverUse ?F .
  ?CITY rdf:type dbpedia:City ;
    strdf:geometry ?CGEO .
  ?BA rdf:type noa:BurntArea ;
    strdf:geometry ?BAGEO .
  FILTER (strdf:Intersect(?R GEO, ?BAGEO) &&
   strdf:distance(?BAGEO, ?CGEO) < 0.02) }
```
We define a SPARQL extension function for each function defined in the OpenGIS Simple Features Access standard.

- **Basic functions**
  - Get a property of a geometry (e.g., strdf:srid)
  - Get the desired representation of a geometry (e.g., strdf:AsText)
  - Test whether a certain condition holds (e.g., strdf:IsEmpty, strdf:IsSimple)

- **Functions for testing topological spatial relationships** (e.g., strdf:equals, strdf:intersects)

- **Spatial analysis functions**
  - Construct new geometric objects from existing geometric objects (e.g., strdf:buffer, strdf:intersection, strdf:convexHull)
  - Spatial metric functions (e.g., strdf:distance, strdf:area)

- **We define spatial aggregate functions** (e.g., strdf:union, strdf:extent)
stSPARQL and OGC standards

• **Spatial terms**
  - Constants (e.g., "POLYGON((38.16 23.7, ...)) " ^^strdf:WKT)
  - Variables (e.g., ?GEO)
  - Results of set operations (e.g., strdf:intersection, strdf:union)
  - Results of geometric operations (e.g., strdf:boundary, strdf:buffer)

• **Select clause**
  - Construction of new geometries (e.g., strdf:buffer(?geo, 0.1))
  - Spatial aggregate functions (e.g., strdf:extent(?geo))
  - Metric functions (e.g., strdf:area(?geo))

• **Filter clause**
  - Functions for testing topological spatial relationships between spatial terms (e.g., strdf:contains(?G1, strdf:union(?G2, ?G3)))
  - Numeric expressions involving spatial metric functions (e.g., strdf:area(?G1)<=2*strdf:area(?G2)+1)
  - Boolean combinations

• **Having clause**
  - Boolean expressions involving spatial aggregate functions and spatial metric functions or functions testing for topological relationships between spatial terms (e.g., strdf:area(strdf:union(?geo))>1)

• **Updates**

---

*Similar to the OGC standard GeoSPARQL*

05/11/12
Strabon: A Scalable Geospatial RDF Store

stRDF graphs

stSPARQL/GeoSPARQL queries

Query Engine
- Parser
- Optimizer
- Evaluator
- Transaction Manager

Storage Manager
- Repository
- SAIL
- RDBMS

GeneralDB

PostGIS

05/11/12
Linked Data used in Fire Monitoring

- Hotspots detected by the National Observatory of Athens (NOA) using MSG-SEVIRI acquisitions
- Hotspots detected by the Fire Information for Resource Management System (FIRMS) using TERRA/AQUA-MODIS acquisitions
- Administrative Regions of Greece
- Corine Land Use / Land Cover Nomenclature
- LinkedGeoData
- GeoNames
Linked Open Data (1/5)

- Hotspots
Linked Open Data (2/5)

- Greek Administrative Geography
• Corine Land Use / Land Cover
Linked Open Data (4/5)

- LinkedGeoData
Linked Open Data (5/5)

- Geonames
Discovering raw data and products

- Retrieve shapefiles that contain acquisitions taken between 20:00 and 20:30 of August 21, 2010 and performed by sensor MSG2

SELECT ?filename
WHERE {
  ?file noa:hasFilename ?filename .
  ?file noa:hasAcquisitionTime ?sensingTime .
  FILTER( str(?sensingTime) >= "2007-08-26T12:00:00" ) .
  FILTER( str(?sensingTime) <= "2007-08-26T12:30:00" ) .
  ?file noa:isDerivedFromSensor ?sensor .
  FILTER( str(?sensor) = "MSG2" ) .
  FILTER( str(?chain) = "StaticThresholds" ) .
}
Retrieve shapefiles that contain acquisitions taken between 20:00 and 20:30 of August 21, 2010 and performed by sensor MSG1_RSS

<table>
<thead>
<tr>
<th>?filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSG2_07-08-26_12:00_StaticThresholds.shp</td>
</tr>
<tr>
<td>MSG2_07-08-26_12:15_StaticThresholds.shp</td>
</tr>
<tr>
<td>MSG2_07-08-26_12:30_StaticThresholds.shp</td>
</tr>
</tbody>
</table>
• Get all hotspots detected in Peloponnesus at 24/08/2007.

WHERE { ?h rdf:type noa:Hotspot; noa:hasGeometry ?hGeo; noa:hasAcquisitionTime ?hAcqTime; noa:hasConfidence ?hConfidence; noa:isProducedBy ?hProvider; noa:hasConfirmation ?hConfirmation; noa:isDerivedFromSensor ?hSensor; noa:isDerivedFromSatellite ?hSatellite; noa:producedFromProcessingChain ?hChain . FILTER(str(?hChain) = "StaticThresholds").
FILTER(?hAcqTime = "2007-08-24T14:45:00"^^xsd:dateTime). FILTER(strdf:contains("POLYGON((21.027 38.36, 23.77 38.36, 23.77 36.05, 21.027 36.05, 21.027 38.36))"^^ strdf:WKT, ?hGeo)).

05/11/12
Improve product accuracy

- Delete part of the polygons that lie in sea.
• Get all coniferous forests in Peloponnese

```turtle
SELECT ?a ?aGeo
WHERE{
  ?a rdf:type clc:Area;
  clc:hasLandUse ?aLandUse;
  noa:hasGeometry ?aGeo.
  ?aLandUse rdf:type ?aLandUseType.
  FILTER(?aLandUseType = clc:ConiferousForest).
  FILTER(strdf:contains("POLYGON((21.027 38.36, 23.77 38.36, 23.77 36.05, 21.027 36.05, 21.027 38.36))"^^strdf:WKT, ?aGeo)).
}
```
Creating a map (3/4)

- Get all municipalities of Peloponnese

```c
SELECT ?d ?dGeo
WHERE {
  ?d rdf:type gag:Dhmos;
  FILTER(strdf:contains("POLYGON((21.027 38.36, 23.77 38.36, 23.77 36.05,
  21.027 36.05, 21.027 38.36))"^^strdf:WKT, ?dGeo)).
}
```

05/11/12
• Get all primary roads in Pelloponnese

```sql
SELECT ?r ?rGeo
WHERE{?r  a ?rType ;
    noa:hasGeometry ?rGeo .FILTER(?rType = lgdo:Primary) .FILTER(strdf:contains("POLYGON((21.027 38.36, 23.77 38.36, 23.77 36.05, 21.027 36.05, 21.027 38.36))"^^strdf:WKT, ?rGeo) ).
```
Final map
Final map
Get roads that are threatened by fires of 2011 according to effis.

```
SELECT ?r (strdf:intersection(?rGeo, strdf:buffer(?hGeo, 0.0114)) AS ?rSegment)
WHERE {
  ?h a noa:Hotspot;
  noa:hasGeometry ?hGeo;
  noa:hasAcquisitionTime ?hAcqTime;
  noa:isProducedBy ?hProvider.
  FILTER (str(?hAcqTime) > "2011-08-01:00:00:00" && str(?hAcqTime) < "2011-11-31:23:59:00").
  FILTER ( ?hProvider = noa:effis ).
  ?r a lgdo:Residential;
  noa:hasGeometry ?rGeo.
  FILTER (strdf:anyInteract(?rGeo, strdf:buffer(?hGeo, 0.0114))).
}
```
More complex examples (2/2)

Get all hospitals that may be threatened from fires in the summer of 2007 according to the FIRMS/MODIS hotspots.

```
SELECT DISTINCT ?nGeo ?nLabel
WHERE { ?d a gag:Dhmos;
?h a noa:Hotspot;
  noa:hasGeometry ?hGeo; noa:hasAcquisitionTime ?hAcqTime;
  noa:isProducedBy ?hProvider.
?n a lgdo:Hospital;
  geo:geometry ?nGeo;
  rdfs:label ?nLabel.
FILTER( strdf:anyInteract(?nGeo, strdf:buffer(?hGeo, 0.025)))
}
```

05/11/12
The Virtual Observatory for TerraSAR-X Data

• Develop a VO that goes **beyond the current EOWEB portal of DLR** to allow queries that take into account image annotations capturing the knowledge available in the images and related GIS data.

• Develop **rapid mapping applications** on top of the VO:
  - Flood monitoring and support
  - Evaluating infrastructure damages after earthquakes
  - Evaluating tsunami effects

• Other rapid mapping applications may be studied if they can increase the visibility of TELEIOS.
## Feature Extraction Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Input parameter</th>
<th>Number of features</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GLCM:</strong> Grey-Level Co-occurrence Matrix</td>
<td>Orientation (1 - 4)</td>
<td>48</td>
</tr>
<tr>
<td><strong>NSFT:</strong> Nonlinear Short Time Fourier T</td>
<td>-</td>
<td>6</td>
</tr>
<tr>
<td><strong>GAFS:</strong> Gabor Filters</td>
<td>scaleGaussian (2 or 4) Orientation (2 or 6)</td>
<td>8 (2 and 2) 48 (4 and 6)</td>
</tr>
<tr>
<td><strong>QMFS:</strong> Quadrature Mirror Filters</td>
<td>nNbLevels (1 or 2)</td>
<td>8 (1) 14 (2)</td>
</tr>
<tr>
<td><strong>LWZ:</strong> Dictionary based compression</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Semantic Annotation

Forest
- Forest mixed

Water

Urban built-up
- Industrial
- Port
- Residential area

Transportation
- Bridge
- Sea
- Channel
- Train lines
RDF Encoding

dlr:Product_1 rdf:type dlr:Product ;
dlr:hasImage dlr:Image_1.tif ;
dlr:hasName "TSX1_SAR"^^xsd:string ;
dlr:hasXMLfilename "TSX1_SAR.xml"^^xsd:string .

dlr:Image_1.tif rdf:type dlr:Image ;
dlr:hasName "IMAGE_HH_SRA_spot_047.tif"^^xsd:string ;
dlr:consistsOf dlr:Patch_1.jpg ;
dlr:hasGeometry "POLYGON((12 45,13 45,13 46,12 46,
12 45))"^^strdf:WKT .

dlr:Patch_1.jpg rdf:type dlr:Patch ;
dlr:hasName "Patch_200_0_0.jpg"^^xsd:string ;
dlr:hasSize "200"^^xsd:int ;
dlr:hasIndexI "0"^^xsd:int ;
dlr:hasIndexJ "0"^^xsd:int ;
dlr:hasGeometry "POLYGON (12 44, 13 44, 13 45, 12 45,
12 44))"^^strdf:WKT ;
dlr:hasGAFS_vector dlr:GAFS_2_2_1 ;
dlr:hasLabel dlr:Label_1 .

dlr:Label_1 rdf:type dlr:Label ;
dlr:correspondsTo dlr:Bridge .

dlr:GAFS_2_2_1 rdf:type dlr:GAFS_Vector ;
dlr:hasFeatureVectorValues dlr:GAFS_2_2_1_values .
Find all the patches containing water limited in the north of a port, at a distance of no more than 200 meters.

```
SELECT ?p1 ?g1
WHERE { ?p1 rdf:type dlr:Patch ;
                dlr:hasGeometry ?g1 ;
                dlr:hasLabel ?l1 .
                ?l1 rdf:type dlr:Label ;
                dlr:correspondsTo dlr:Water .

            ?p2 rdf:type dlr:Patch ;
                dlr:hasGeometry ?g2 ;
                dlr:hasLabel ?l2 .
                ?l2 dlr:correspondsTo dlr:Port ;
                rdf:type dlr:Label .

            FILTER(strdf:above(?g1,?g2)) .
            FILTER(strdf:contains(strdf:buffer(?g2,0.005),?g1))}
```

Querying on semantic and spatial information

05/11/12
Querying on semantic information and feature values

- Find all patches that correspond to a bridge, according to the 1st feature value of the Gabor algorithm with 4 scales and 6 orientations.
The Framework RDFi

- Extension of RDF with **incomplete information**
- New kind of literals (**e-literals**) for each datatype
  - Property values that *exist* but are *unknown* or *partially known*
- **Partial knowledge**: captured by constraints (appropriate constraint language \( L \))
- RDF graphs **extended** to RDFi databases: pair \((G, \varphi)\)
  - \(G\): RDF graph with e-literals
  - \(\varphi\): quantifier-free formula of \( L \)
- Formal semantics for RDFi and SPARQL query evaluation
- **Representation System**: CONSTRUCT with AUF graph patterns
- **Certain Answer**: semantics, algorithms, computational complexity when \( L \) is a language of spatial topological constraints
- Implementation in the context of Strabon has started with \( L \) being PCL (topological constraints between variables and polygon constants)
hotspot1 type Hotspot.
fire1 type Fire.
hotspot1 correspondsTo fire1.
fire1 occurredIn _R1.
_R1 TPP "x ≥ 6 ∧ x ≤ 23 ∧ y ≥ 8 ∧ y ≤ 19"
RDF\textsuperscript{i} Database: An example

hotspot1 type Hotspot .
fire1 type Fire .
hotspot1 correspondsTo fire1 .
fire1 occurredIn _R1 .
_R1 NTPP "x\geq 6 \land x\leq 23 \land y\geq 8 \land y\leq 19"
hotspot1 type Hotspot .
fire1 type Fire .
hotspot1 correspondsTo fire1 .
fire1 occurredIn _R1 .
_R1 NTPP "x≥6 ∧ x≤23 ∧ y≥8 ∧ y≤19"
Example Query

- Find all fires that are inside region Q1
Example Query

- Find all fires that are inside region Q1
• In stSPARQL this query would be expressed as

```
SELECT ?F
WHERE {
    ?F type Fire .
    ?F occurredIn ?R .

    FILTER (NTPP(?R, "x≥10 ∧ x≤21 ∧ y≥12 ∧ y≤17"))
}
```
• What is the answer to the previous query?
What is the answer to the previous query?
• What is the answer to the previous query?

• It is not certain that fire1 is inside Q1.
• What is the answer to the previous query?

• It is not certain that fire1 is inside Q1.

• The answer should be conditional.
<table>
<thead>
<tr>
<th>?F</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>fire1</td>
<td>_R1 NTPP “x≥10 ∧ x≤21 ∧ y≥12 ∧ y≤17”</td>
</tr>
</tbody>
</table>

- The answer reads as: “fire1 is in the answer only if the respective region where it occurs is inside rectangle Q1”

- Answers reminiscent of c-tables [Imielinski-Lipski84]
Example Query

• Find all fires that are certainly inside Q2
Example Query

• In stSPARQL this query would be expressed as

```
CERTAIN SELECT ?F
WHERE {
  ?F type Fire .
  ?F occurredIn ?R .
  FILTER (NTPP(?R, “x≥2 ∧ x≤28 ∧ y≥4 ∧ y≤22”))
}
```
Answer to Example Query

- What is the answer to the previous query?
Answer to Example Query

• What is the answer to the previous query?
• What is the answer to the previous query?

Now it is **certain** that fire1 is in Q2. **Why?**
• What is the answer to the previous query?

• Now it is certain that fire1 is in Q2. Why?

• Because fire1 occurs in a region inside P (qualitative) which is known (quantitative) to be inside Q2 (the geometries of P and Q2 are known).
Conclusions

- **stRDF/stSPARQL**
  - Model and query language for the representation and querying of geospatial data

- **Strabon**
  - Scalable Geospatial RDF Store

- **Linked Open Data**
  - Add value to final product
Vielen Dank!

- How about building a toy application with your spatial data?
  - Strabon
  - NOA Application Demo
  - TELEIOS EU Project
    [http://www.earthobservatory.eu/](http://www.earthobservatory.eu/)